# Tekton Build Credentials Exchange via Vault

Mit diesem PoC wollen wir die Möglichkeit skizzieren, die Secrets für den Zugriff von Tekton auf Bitbucket und Artifactory transparent im jeweiligen Tekton Namespace anzulegen, ohne dass der Abnehmer damit in Berührung kommen muss.

Für erste Tests wurde ein Vault Namespace auf NonProd für ESTA registriert: <https://vault-nonprod.sbb.ch/ui/vault/secrets/kv/list/esta/openshift/>

Lese-Zugriff sollte für alle Mitglieder von DG\_RBT\_ESTA funktionieren, Schreib-Zugriff ist für DG\_RBT\_ESTA\_ADMIN registriert.  
Für das Login in Vault muss man zuerst die Mitgliedschaft in den Gruppen DG\_RBT\_HASHICORPVAULT\_USER\_NONPROD resp. DG\_RBT\_HASHICORPVAULT\_USER\_PROD beantragen (via [rbt.sbb.ch](https://rbt.sbb.ch/)).

Die M2M Authentifizierung geschieht über einen spezifischen Service Account im jeweiligen Namespace (siehe [JWT Authentication](file:///C:\display\IAM\JWT+Authentication)). Für den Test auf clew01d / esta-tekton-predev haben wir einen Service Account [vault-auth-tekton](https://console-openshift-console.apps.clew01d.sbb-aws-dev.net/k8s/ns/esta-tekton-predev/serviceaccounts/vault-auth-tekton) erstellt und registriert.

Das Secret in Vault hat die folgende Struktur (JSON):

**esta/openshift/esta-tekton-predev/bitucket-user-credentials**

{

"username": "esta-build-clew",

"password": "xxxxxx",

"ssh-privatekey": "-----BEGIN OPENSSH PRIVATE KEY-----\n....\n-----END OPENSSH PRIVATE KEY-----\n"

}

Nun können mit dem folgenden Ressourcen die Bitbucket User Credentials unter dem Pfad **esta/openshift/esta-tekton-predev/bitucket-user-credentials** in den Namespace esta-tekton-predev synchronisiert werden:

 Quelle erweitern

apiVersion: external-secrets.io/v1beta1

kind: SecretStore

metadata:

name: vault-secretstore-nonprod

namespace: esta-tekton-predev

spec:

provider:

vault:

auth:

jwt:

kubernetesServiceAccountToken:

audiences:

- 'https://kubernetes.default.svc'

expirationSeconds: 600

serviceAccountRef:

name: vault-auth-tekton

path: aws\_clew01d

role: default

path: kv

server: 'https://vault-nonprod.sbb.ch/'

version: v2

---

apiVersion: external-secrets.io/v1beta1

kind: ExternalSecret

metadata:

name: esta-tekton-basicauth-git-sync

namespace: esta-tekton-predev

spec:

data:

- secretKey: username

remoteRef:

key: esta/openshift/esta-tekton-predev/bitucket-user-credentials

property: username

- secretKey: password

remoteRef:

key: esta/openshift/esta-tekton-predev/bitucket-user-credentials

property: password

refreshInterval: 30s

secretStoreRef:

kind: SecretStore

name: vault-secretstore-nonprod

target:

creationPolicy: Owner

deletionPolicy: Retain

name: esta-tekton-basicauth-git-vault

template:

type: kubernetes.io/basic-auth

metadata:

annotations:

tekton.dev/docker-0: 'https://\*.docker.bin.sbb.ch'

tekton.dev/docker-1: 'https://docker.bin.sbb.ch'

data:

username: "{{ .username }}"

password: "{{ .password }}"

---

apiVersion: external-secrets.io/v1beta1

kind: ExternalSecret

metadata:

name: esta-tekton-ssh-git-sync

namespace: esta-tekton-predev

spec:

data:

- secretKey: privatekey

remoteRef:

key: esta/openshift/esta-tekton-predev/bitucket-user-credentials

property: ssh-privatekey

refreshInterval: 30s

secretStoreRef:

kind: SecretStore

name: vault-secretstore-nonprod

target:

creationPolicy: Owner

deletionPolicy: Retain

name: esta-tekton-ssh-git-vault

template:

type: kubernetes.io/ssh-auth

metadata:

annotations:

tekton.dev/git-0: code.sbb.ch

tekton.dev/git-1: codessh.sbb.ch

data:

known\_hosts: |

[code.sbb.ch]:7999 ssh-rsa xxxxx

[codessh.sbb.ch]:7999 ssh-rsa xxxxx

[10.241.17.8]:7999 ssh-rsa xxxxx

ssh-privatekey: "{{ .privatekey }}"

## Tekton Namespace Setup

Damit beim Bestellen eines Tekton Namespace die Credentials autom. via Vault in den Namespace gelangen sind die folgenden Schritte notwendig:

* Bitbucket User mit Credentials anlegen und in Vault speichern (→ WZU Backend)
* OpenShift Namespace anlegen (→ CON Backend)
* Service Account in OpenShift anlegen (→ ??)
* Service Account für JWT Auth in Vault registrieren (→ Vault Backend)
* SecretStore und ExternalSecret Ressources anlegen (→ esta-tekton-pipeline-config Helm Chart)